Experience Report

Applying Ada, Java, and CORBA for Making a Command and Control Information System Platform Independent

Gerhard Bühler
Research Establishment for Applied Sciences
Research Institute for Communication, Information Processing, and Ergonomics
Neuenahrer Straße 20
D-53343 Wachtberg-Werthhoven
Germany
Telephone number: (+49) 228/9435 376
Fax number: (+49) 228/9435 685
E-mail address: buehler@fgan.de

Heinz Faßbender[1]
Research Establishment for Applied Sciences
Research Institute for Communication, Information Processing, and Ergonomics
Neuenahrer Straße 20
D-53343 Wachtberg-Werthhoven
Germany
Telephone number: (+49) 228/9435 640
Fax number: (+49) 228/9435 685
E-mail address: fassbender@fgan.de

Abstract

We describe the concepts and experiences we have made in an ongoing project by applying the middleware standard CORBA and the internet programming language Java in redesigning an experimental command and control information system which is almost written in Ada, for making the user able to use the system from any platform which is connected to the world wide web and which consists at least of a world wide web browser which, in its turn, is able to run Java applets. Furthermore, we develop a general strategy by which an arbitrary platform dependent C2 information system or more generally a legacy system can be redesigned in such a way that it can be used totally platform independently where only a very small amount of system resources is required.

[1] Corresponding Author

1. The Experimental Command and Control Information System EIGER

In our department *Command and Control Information Systems* of our institute an experimental distributed command and control information system called EIGER, has been developed which can support the work of head quarters of the German army. Nearly 90% of the system have been coded in Ada 83. The other 10% have been coded in C for realizing the communication of the system's components. For our current work, EIGER has been changed so far that it can be compiled by an Ada 95 compiler which ensures that more elegant and more time efficient mechanisms can be integrated into the system. We use the Object Ada system from Aonix as development environment. The current version runs on Sun workstations with Solaris. Since the kernel of the system can be compiled and linked by the Object Ada system on Windows NT, we have realized the first step of transporting the complete system to Windows NT.

Unfortunately, the architecture of the existing version *is totally platform dependent*, because EIGER can be only used on Solaris which extremely restricts its applicability. This grievance should be avoided by the new version. And in the following we will discuss, how we will redesign the current system such that the system *becomes totally platform independent.* But, before we are able to do this, we have to discuss some basic aspects of the current structure of EIGER (cf. [Bühler, 1998][2] for an intensive description of EIGER).

EIGER consists of a finite amount of subsystems the communication of which is realized by a communication system. The implementation of the communication system is not important for understanding the redesigning mechanisms. For this purpose, we only have to understand, how a subsystem is structured (cf. Figure 1).

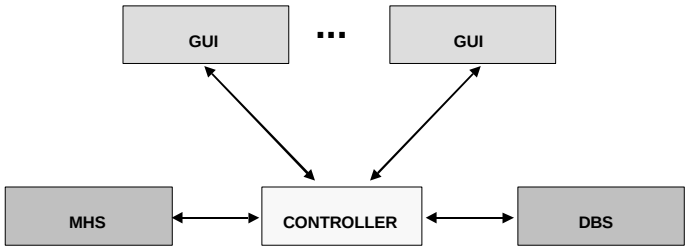

Figure 1. Structure of a subsystem

With great foresight a subsystem has been structured as a multi tier process system. Without the distribution and strong modularization of the subsystem the modification would be not as efficient and simple realizable as it will be presented in the following.

The central unit of a subsystem is the *controller* which controls the computations and communications of the other components. This component is completely implemented in Ada83. The controller is connected to a *relational data base system* DBS which stores the data of the organization corresponding to the underlying data model, to an *X.400 message handling system* MHS, and to one or more *graphical user interfaces* GUI which are produced by OSF/Motif.

2. Concepts for EIGER's Redesign

We integrate the following two concepts into the implementation of a subsystem for getting the desired platform independence:

---

[2] [Bühler, 1998] Gerhard Bühler. *Einsatz von Ada im Experimentellen Führungsinformations-system EIGER*. in Workshop „Entwicklung von Software-Systemen mit Ada", Bremen, Germany, Ada Germany, 1998.

A GUI shall be implemented as Java applet. Then it (and thus, the complete EIGER system) can be applied on every computer system which is connected to the internet and which includes at least a web browser which in its turn is able to run Java applets.

The communication between the controller and the GUI shall be realized by a CORBA connection. This also supports the platform independent use of the system, since CORBA is a system independent standard which can be implemented on nearly every system for nearly every operating system and nearly every programming language.

Integrating these two concepts leads to the following new structure of a subsystem which is illustrated in Figure 2. Remark that we omit the connections to the data base system and to the message handling system, because they are not changed in this first restructuring step.

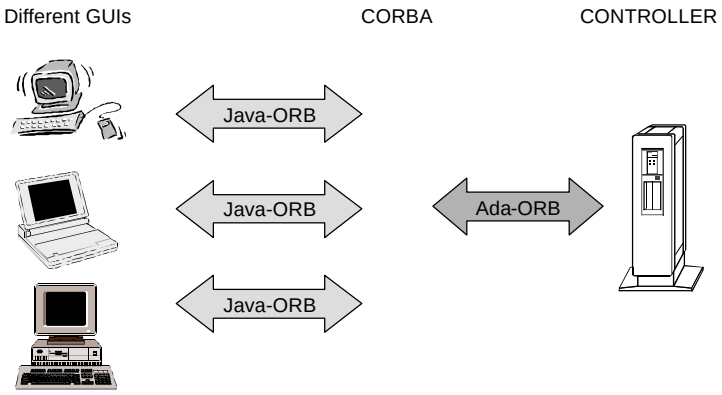

Figure 2. Redesign of a subsystem

3. Problems and Solutions

Finally, we discuss the problems which have to be solved for implementing the structure which is illustrated in Figure 2, and present our solutions. In addition to the presented problems we have had to integrate an efficient call back mechanism, since the CORBA connection between the GUI and the controller has to be realized bidirectionally.

3.1 *Reimplementation of GUIs as Java Applets*

This problem is completely solved by a new implementation of the GUIs as Java applets under the system JBuilder2 from Inprise (Borland & Visigenic).

3.2 *Connection of the GUI by a Java-ORB*

As Java-ORB we use the Visibroker from Inprise which is integrated into the JBuilder2 Client and Server Version. We have not had any problem for communications of Java applications. But some problems arose for communications of Java applets. Nevertheless, by using the newest version of Netscape these problems have also been solved.

3.3 *Connection of the Controller by an Ada-ORB*

We use the ORBADA system from Top Graph'X as Ada-ORB which works well after some corrections in a previous version of ORBADA. The connection of the Ada-ORB to the controller is very tricky. It will be explained, if we are allowed to give a talk at the conference.

3.4 *Communication between Java-ORB and Ada-ORB*

The applied CORBA systems work well together. So, we can state that they are CORBA conform at least in the checked services.